Hindawi
Journal of Sensors
Volume 2022, Article ID 2267107, 20 pages
https://doi.org/10.1155/2022/2267107

Research Article

On Study of 1D Depth Scans as an Alternative Feature for Human Pose Detection in a Sensor Network

Maryamsadat Rasoulidanesh and Shahram Payandeh

Networked Robotics and Sensing Laboratory, School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6

Correspondence should be addressed to Maryamsadat Rasoulidanesh; rasoulid@sfu.ca

Received 16 April 2022; Revised 18 June 2022; Accepted 25 June 2022; Published 13 August 2022

Academic Editor: Yunchao Tang

Inspired by the notion of swarm robotics, sensing, and minimalism, in this paper, we study and analyze how a collection of only 1D depth scans can be used as a part of the minimum feature for human body detection and its segmentation in a point cloud. In relation to the traditional approaches which require a complete point cloud model representation for skeleton model reconstruction, our proposed approach offers a lower computation and power consumption, especially in sensor and robotic networks. Our main objective is to investigate if the reduced number of training data through a collection of 1D scans of a subject is related to the rate of recognition and if it can be used to accurately detect the human body and its posture. The method takes advantage of the frequency components of the depth images (here, we refer to it as a 1D scan). To coordinate a collection of these 1D scans obtained through a sensor network, we also proposed a sensor scheduling framework. The framework is evaluated using two stationary depth sensors and a mobile depth sensor. The performance of our method was analyzed through movements and posture details of a subject having two relative orientations with respect to the sensors with two classes of postures, namely, walking and standing. The novelty of the paper can be summarized in 3 main points. Firstly, unlike deep learning methods, our approach would require a smaller dataset for training. Secondly, our case studies show that the method uses very limited training dataset and still can detect the unseen situation and reasonably estimate the orientation and detail of the posture. Finally, we propose an online scheduler to improve the energy efficiency of the network sensor and minimize the number of sensors required for surveillance monitoring by employing a mobile sensor to recover the occluded views of the stationary sensors. We showed that with the training data captured on 1 m from the camera, the algorithm can detect the detailed posture of the subject from 1, 2, 3, and 4 meters away from the sensor during the walking and standing with average accuracy of 93% and for different orientation with respect to the sensor by 71% accuracy.

1. Introduction

Analysis of human posture and recognition of its movements and action are some of the key elements in various fields such as health, entertainment, and security. Despite the current advances, the field still faces many challenges due to variation in human postures, its appearances, the partial occlusion, the presence of complex background, and variation in illumination conditions. Similar general recognition challenges have been identified in other fields such as agriculture [1, 2], construction [3], and manufacturing [4].

Traditionally, RGB cameras are employed to detect the presence and recognition of the human body. Recently, various tools and approaches from signal processing, deep learning, and artificial intelligence to sensor networks and robotics have shown great promise. However, these approaches usually rely on a large amount of data for their training and implementation [5–13]. The introduction of time-of-flight sensor, i.e., depth sensor, has also contributed significantly to associating sensed information in the construction of the spatial point cloud information from the scene. The key advantages of the depth sensors are their

robustness against environmental effects such as illumination changes or color inconsistency [14–22].

Inspired by the notion of swarm intelligence [23], in this paper, we explore and study a novel notion of RoI (region of Interest) to segment the human body in each frame associated with the sensor and robotic network. The proposed RoI model is inspired by the human perception system for reducing the effect of unwanted distractors in the scene [24]. These models gained broad attraction due to their ability to enable the tracking system to focus on the possible positions of the tracked target. In this paper, a method is proposed where changes are detected using the sampled point cloud representation within RoI. In order to reduce the associated computational complexity, we utilize selective point cloud sampling. We refer to such a sampling approach as 1D scans. The employment of 1D scans allows minimization of the sensed information through the sensor and robotic network.

However, since the depth frame is noisy and contaminated with outliers and flying pixels, a series of preprocessing steps need to be carried. FFT profile of 1D scans has been used within the framework of SVM classifier to identify the real changes in the scanned profile. If a change has been detected on any of the reference 1D scans, RoI windows are anchored within the location of changes in that 1D scan. The size of the extracted window can vary depending on the position and size of the detected change and its distance with respect to the depth sensors. As such, the proposed methodology can be extended to detect more than one person. After reliably detecting the position of the subject with respect to the sensor, we utilize the predefined number of scans to estimate subject orientation with respect to the sensor, and our goal is to use the minimum number of scans to accurately estimate the orientation of the subject which consequently decreases the computational complexity that is aligned with minimal sensing. Finally, we showed that our method can be employed in calibration-free scenarios by multiple sensors in a sensor network to extend the field of view of the monitoring area. In addition, the proposed sensor network contains a robot sensor to freely move around in the monitoring environment and cover the areas out of the field of view of the stationary sensors.

The remaining of the paper is organized as follows. In Section 2, a review of recent related literature is presented. Section 3 presents an overview of the change detection method and the extraction of RoI window. Section 4 presents the structure of the proposed classifier to detect human posture in a sensor and robotic network. Multisensor scheduling and using robot sensors are explained in Section 5. The experimental results associated with the effectiveness of using the minimum number of training data for detection both in the view of stationary sensors and mobile sensors are discussed in Section 6 along with details of the real-time implementation of the paper. Discussions and conclusions are presented in Section 7.

2. Related Work

The emergence of depth sensors allows the incorporation of synchronized spatial information of the scene which can improve the detection and segmentation tasks. In the literature, three main approaches have been proposed to incorporate depth data. The first approach uses depth along with RGB as a piece of auxiliary information [25, 26]. In this approach, the main data are extracted from the RGB frames, and then the depth information is utilized to improve and complement the data through RGB images. In the second approach, both depth and RGB frames are considered to extract the relevant features [27]. In [28], the depth data is fused to yolo structure [29] in three different stages (early-stage, mid-stage, and late-stage) and showed that the best results can be achieved by fusing the depth information in the middle layers of yolo. In the third approach, only depth frames are utilized to accomplish the assigned task [30–33]. In [34], it is shown that the efficiency of using depth information can be better than the results of RGB information in some tracking scenarios.

Using deep-learning on RGB-D images is also increasingly becoming popular. [35] presents one of the earliest human pose detections using random forest technique. Many approaches are trying to extract human silhouettes and human poses using various structures of CNN [28, 36, 37]. [33] used only depth frames to locate the head of the subject in the scene and then utilized depth map, multiorder depth template, and height difference map as inputs to be fed into a pretrained CNN. [36] extracted convolutional features on the depth map and then used k -th nearest neighborhood to find the corresponding parts of humans in different depth frames. The skeletal points are mainly extracted from the depth data and then employed for action or pose detections [38–41]. People reidentification is another widely explored area by using depth data [42] as a process in which an individual is tagged and identified in a multisensor field. [43] provided a synthesized dataset for people reidentification including depth and RGB information. Using depth information such as depth similarity is assisting on segmentation tasks in [44]. They used CNN for object segmentation by taking advantage of depth similarity [44]. Many of the above methods used RGB trained convolutional neural networks as the basis of the training. However, the different nature of the depth information suggests the benefit of using different models between RGB and depth information. This also motivates the usage of depth information for feature learning which can also lead to a complementary feature learning for the recognition process.

An important aspect of this paper is to propose a method that can reduce the amount of sensed data for pose estimation and tracking in relation to complete RGB image point cloud for skeleton model reconstruction. The notion of minimal data for tracking using weak detection sensors (those which provide simple crossing information) is discussed in [45]. Minimal sensing is an important issue in many applications including robotics [46], quadrotor flight [47], remote sensing [48], and vision [49]. To address the requirements of minimal sensing, two main strategies have been utilized in this paper. First, only part of the depth map has been used as an input for estimating the pose of the subject. The performance of this approach is experimentally demonstrated which shows the relationship between the number of depth

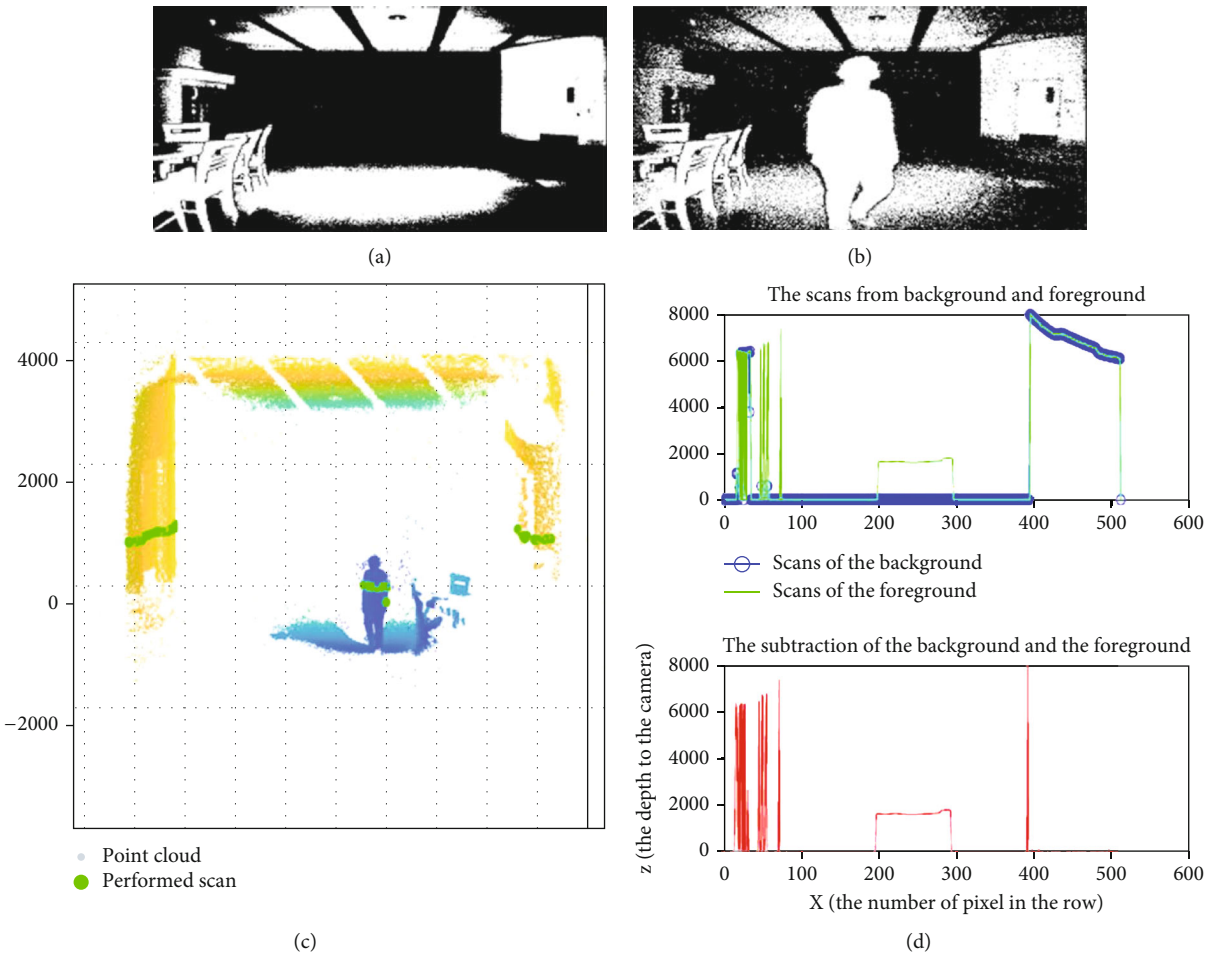

Figure 1: (a) Depth image of the background. (b) Depth image of the foreground. (c) Point cloud of the foreground and the selected scan showed by green points. (d) Selected scan from the background and the foreground and their subtractions.

scans (extracted from depth map) and the estimation accuracy. Second, it is shown that the collected training data at the location of one meter from the sensor plane can result in a good pose estimation accuracy when the subject is placed further than one meter (up to four meters) from the sensor. In the proposed method, we do not extract any skeleton points to be used as a part of pose detection; instead, we are using a collection of 1D scans. To the best of our knowledge, majority of the state of the art in pose detection algorithms are based on extraction of skeleton points ([50, 51]) which requires a more complex steps compared to extracting based on series of 1D scans.

As a part of the adaptive active sensing strategy, we utilized a robot equipped with a depth sensor to dynamically extend the field of view of the monitoring, relative to the position of the stationary sensors. The robot can move to some adaptively defined trajectory based on the prediction of the subject's movements. In [52], the authors used a mobile robot in a supermarket environment to detect the human body and distinguish between two main postures as standing and squatting. They cluster the point clouds to segment the human body. They then divided the segmented point cloud into four main regions to separate the standing from squad postures. They used surface normal associated with patches of the point cloud as features to be fed to 1vs1-SVM classifier for human nonhuman classification and different postures. In [53], a depth sensor mounted on a mobile robot is used to estimate the posture of the subject for lying down, sitting, standing, and bending over.

In this paper, we take advantage of a novel RoI window to improve real-time performance and minimize the necessary information for pose detection and tracking. The proposed method consists of four stages. (1) change detection, (2) RoI extraction, (3) classification, and finally, (4) data fusion from multiple sensors and a robot. The first part of the algorithm is responsible for the basic extraction of any potential movement in the scene followed by steps to remove any noise and outliers leading to deciphering the most reliable changes in the scene. Most importantly, instead of subtracting the complete depth data, we introduce the new method by incorporating a collection of 1D scans (i.e., by taking advantage of one single row of the depth image at a time). This is our minimalization approach to sensing information and can be employed in many applications with limited resources. After the change detection step, the RoI area is built to contain the depth points belonging to the detected object. The frequency feature characteristics of the detected change are then extracted and incorporated into

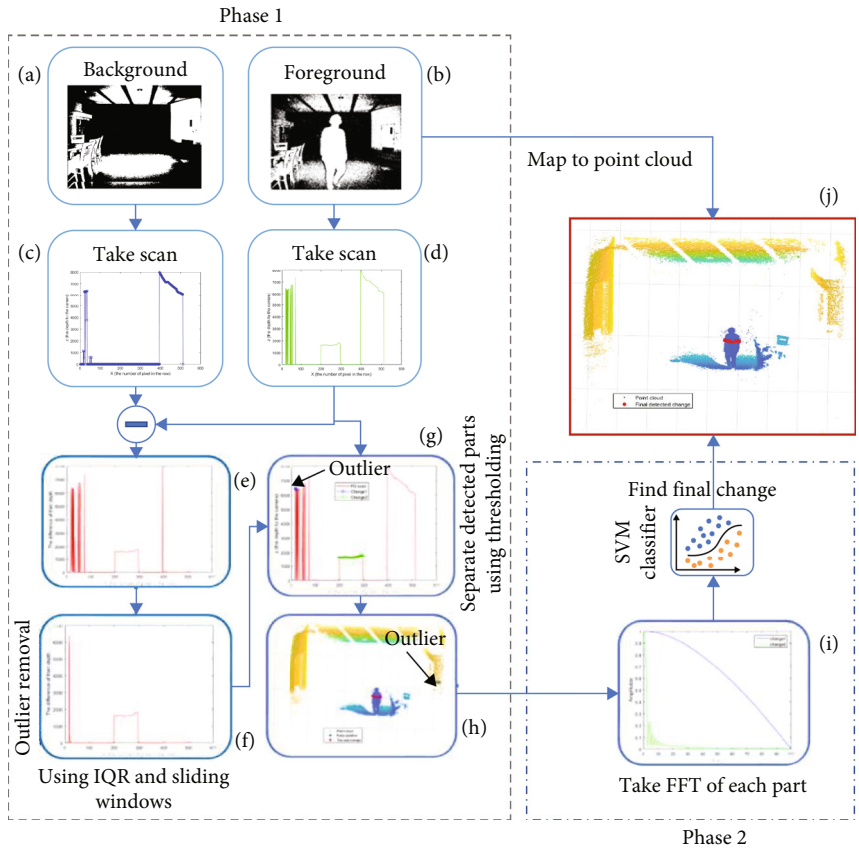

FIGURE 2: Change detection: (a) sample background, (b) sample foreground, (c) 1D scan from background, (d) 1D scan from foreground, (e) the subtraction of foreground and background scan, (f) the subtraction of two scans after outlier removal step, (g) result of clustering to find changes in the foreground scan, (h) corresponding changes on point cloud, (i) frequency of the scan, and (j) result of change detection.

the body detection and posture classifier. These extracted features along with their spatial information are utilized in the network of sensors were the most qualified sensor(s) are assigned to track and monitor the body. Finally, a robot sensor is utilized to find the subject in an uncovered monitoring area with the help of the schedular. This paper provides the method to use a smaller dataset and by using FFT features, we are able to predict the unseen situations (subjects positioned at different locations with respect to the sensors) reasonably accurate. In addition, we have used an schedular to improve the energy efficiency of the system and minimize the number of sensors required to monitor the surveillance environment.

3. Preliminaries: Overview of Change Detection and RoI Extraction

A novel change detection algorithm for event monitoring was previously proposed by the authors using only depth information [54]. In this section, we present an overview of this algorithm with newly added features relevant to the objectives of this paper. This is an important phase of the algorithm which allows the definition of an RoI window for processing only data points to be defined within the RoI which are confined to the tracked subject. This is also a part of the overall minimalization framework by focusing on a part of the frame instead of the whole frame in reducing computational overheads. The detected change is then used as a seed region in order to segment the foreground and further define the RoI. In the following subsections, we first present details of the change detection algorithm and then outline the RoI extraction method.

3.1. Change Detection Algorithm. Initially, we assume that we have a set of depth images which correspond to a fixed background. Ideally, the subtraction of the foreground and the background should result in detecting the changes in the scene. However, depth images are usually very noisy and unstable. In addition, imperfections of depth images such as missing points result in ambiguity in the measurements. Denoising and smoothing of the depth images were the main topics of many past works such as [55, 56]. However, these methods are usually time-consuming and not practical for real-time applications. In work [54], we showed that how learning the features which can include distance and angle from sensor center can improve the noise removal process. This information can be used to compute the amount of contribution of each pixel by estimating the chance of a pixel affected by noise using a statically model of the sensor placed at different positions with respect to the monitoring environment. In this paper, we utilize a similar concept but instead, we train a model to learn the

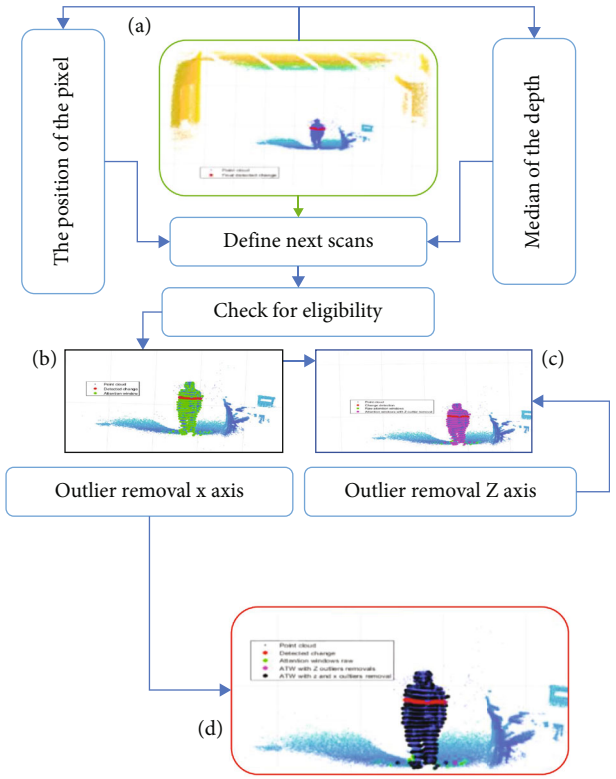

FIGURE 3: Extraction of RoI window based on the detected changes shown in Figure 2(h): (a) the detected change, (b) extending the results, (c) outlier removal on z axis, and (d) outlier removal.

behavior of the noise and use that it to mitigate its impact. We focus on finding changes in a single selected 1D scan in order to associate them with changes in the scene. A 1D scan contains only 1 row of the depth image, which implies that it has a lower dimension than the original depth image. Let $D = D(i, j)$ (for example, for Microsoft Kinect V2, $i = \{1, 512\}$, $j = \{1{,}422\}$) be the depth image (i.e., [512,424]); then, a 1D scan which is a row of the depth image can be defined as $S_a(j)$ where $j = \{1{,}422\}$ and $1 < a < 512$ are the scan indicator.

Figures 1(a) and 1(b) show the depth frame of background and foreground, respectively. Figure 1(c) illustrates the corresponding point cloud to the foreground where the green points belong to the selected scan. Figure 1(d) shows the scans obtained from the background and the foreground (top image), along with their subtraction (bottom image). As can be seen, the simple subtraction of the two scans results in multiple false change detections due to the presence of numerous noise and outliers. In order to eliminate the noise, we proposed previously to determine the weight of each point that contributes to the detected changes based on its position with respect to the depth sensor. We estimate changes in each pixel by formulating a noise band corresponding to the pixel to be affected by noise. In this paper, we proposed to learn the behavior of noise and train a SVM model to classify noise and no noise for detecting changes.

Figure 2 shows the steps which are followed in order to reliably detect any changes in the scene and remove the faulty detections. In the first step, a 1D scan of both background (Figure 2(a)) and foreground (Figure 2(b)) is obtained which are shown in Figures 2(c) and 2(d)), respectively. Then, the subtraction of these scans is utilized to identify the candidates of the potential change (shown in Figure 2(e)). To decrease the effect of missing points and outliers, we perform a sliding window IQR (interquartile range) outlier removal technique [57] which is explained below.

Let $4 < \omega < 422$ be the size of the sliding window, in a current window $W = S_a(b : b + \omega)$ (where ω is the size of the sliding window, W is the window, and b is the window start position in the scan). In IQR techniques, we will change the value of $W(j)$ to the median M if $W(j) < m - Q_2$ or $W(j) > m + Q_1$ where Q_1 is the upper quadrant of w and Q_2 is the lower quadrant, and m is its median. The result after IQR outlier removal technique is shown in (Figure 2(f)), where some of the falsely detected changes (false positive) are eliminated from the results. In this stage, we need to cluster the changes of the scan to split up each change in the scene. To do so, the rate of variation in depth value is computed along with the direction of 1D scan (if $j = 0$ is the start of the scan, its direction is in the direction of increasing order of the index). Finally, by applying a selected threshold to the magnitude of the result, it is possible to detect each of the independent changes. Figure 2(g) illustrates two detected clusters in blue and green color. However, the detected change in blue is a false positive and should not be accepted as a change (Figure 2(h)). Figure 2(g) shows the position of each cluster in the point cloud where the blue points are false positive.

Finally, by performing fast Fourier transform (FFT) on each cluster, the frequency features are extracted and normalized between zero and 1 (shown in Figure 2(i)). Using these feature vectors as an input to a supported vector machine (SVM) classifier, the pixels that are candidates for change can be classified into two classes of accepted change and nonaccepted. In the training phase, since the depth image is noisy, the nonacceptable changes happened more frequently. Hence, to make the training data balanced (have an approximately equal number of images in each set), we downsampled the nonaccepted changes in the training set. Figure 2(j) illustrates the result of the approach to find the reliable change on the scene.

The number of scan can vary depending on the application and its position can be different for consecutive frames to make sure that the whole changes of the scene will be covered at the end.

3.2. Foreground Segmentation Using RoI Extraction. The position of the changes in the scene can be used as depth cues for RoI extraction using a collection of primary 1D scans instead of the full depth frame. Here, we perform scans defined by $s_{bi} : b_i = a \pm \alpha i$ where a is the position of the primary scan and α is the increment factor between the consecutive scans and is determined based on the estimated depth of the object in the foreground.

Figure 3 illustrates the steps needed in order to extract the RoI following Figure 2(h). Points from any new scans

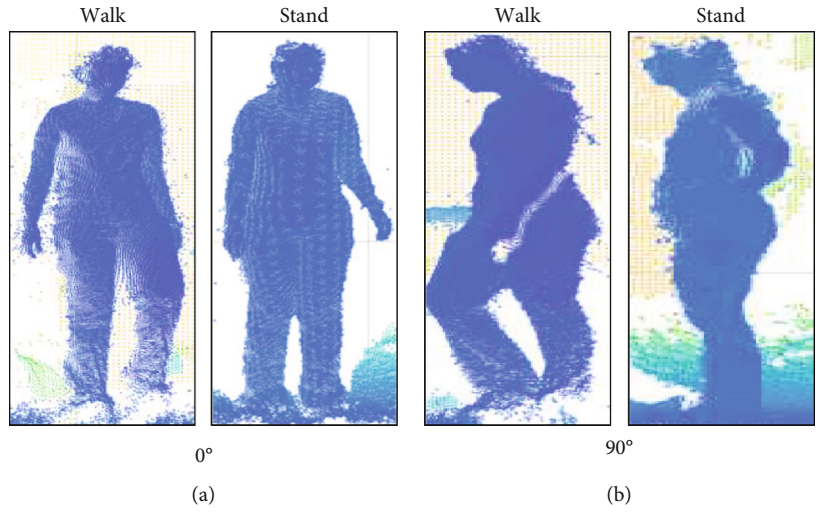

Figure 4: Two classes of main orientation with respect to the depth sensor in two postures of walking and standing.

are not collected when they are determined to have the lower expected value of belonging to the same detected object. For example, if the detected change in the scan s_a is in the range of l_1 and l_2 (determined in the scan the change has been detected), then the RoI in the immediate next scan would be from $[l_1 - \tau, l_2 + \tau]$ where τ is the extending factor that can be determined based on the estimated depth of the object under consideration. The extraction of change in the new scan will be done using the same steps as subsection phase 1 in Figure 2. Detected points (if any) in this new scan are recorded as changes and will be added to the set foreground (Figure 3(b)). Finally, the outlier will be removed from the detected object along all three dimensions. The result of the outlier removal is shown in Figures 3(b)–3(d) which is confined to the human body. Using this approach, other objects (rather than humans) can also be detected by classifying any changes to the human or nonhuman class. This can further be used as a part of human posture estimation.

4. Human Posture Estimation

In this section, we will use the frequency features of 1D scans to detect and further classify body posture. As a part of an illustration and experimental studies, we aim to detect and track the subject in a network of sensors consisting of two stationary depth sensors and one mobile robot equipped with a depth sensor. Two classes of information from each tracked subject are investigated: first, information regarding how to distinguish between the relative locations of the subject with respect to each sensor and the robot, and second, information related to movement details, e.g., the movement of the legs while the subject is walking. Our main objective is to study how minimization of the number of training data of a subject is related to the rate of recognition. Furthermore, we aim to explore how using training data obtained at a single location with respect to a sensor can affect the accuracy of posture estimation throughout the whole monitoring area. The main features utilized in this paper are mainly the frequency profile of the separately performed scans from the subject point clouds. In the following, details associated with the feature extraction are presented.

4.1. Feature Extraction. The purpose of this study is to distinguish various details between classes of body posture. In works [58, 59], we showed that the frequency features can be used to accurately classify the general overall postures of the subject, e.g., sitting, standing, or lying down. However, no further analysis was carried in regard to the level of detail associated with each posture (e.g., the location of the subject with respect to the sensor of the relative pose of limbs). Additionally, it is possible to define various body postures as key postures where one can interpret several intermediates once between any two consecutive key postures. Two main analyses are studied in this work. Firstly, we analyze and distinguish between different relative locations of the body with respect sensor. This distinction is very important in many scenarios such as in determining the direction of movements. An example of this application is to estimate the location of the subject in order to guide the mobile robot sensor toward proximal locations with respect to the subject when the subject is out of the field of view of the stationary sensors. We define two seed orientations of the subject with respect to the sensor as shown in Figure 4(a) which are 0-degree and 90-degree. Naturally, there are many increments of angular orientations but here we are only concerned with two main classes as the main anchor for other angular orientations. In the other words, other orientations can be interpreted through incremental mappings. Our second objective is to distinguish the various poses of a posture. For example, as part of poses associated with a walking sequence, we focus to find out if the subject keeps a leg one in front of the other (posture during walking) or both legs are on the ground and close to each other (posture during standing, Figure 4(b)).

The first step toward our objectives is feature extraction. Figure 5 illustrates our approach to extracting features of each posture. The first step is to detect the subject body (foreground) using the method which was presented in the previous section. Let N_{sf} be the number of scans that

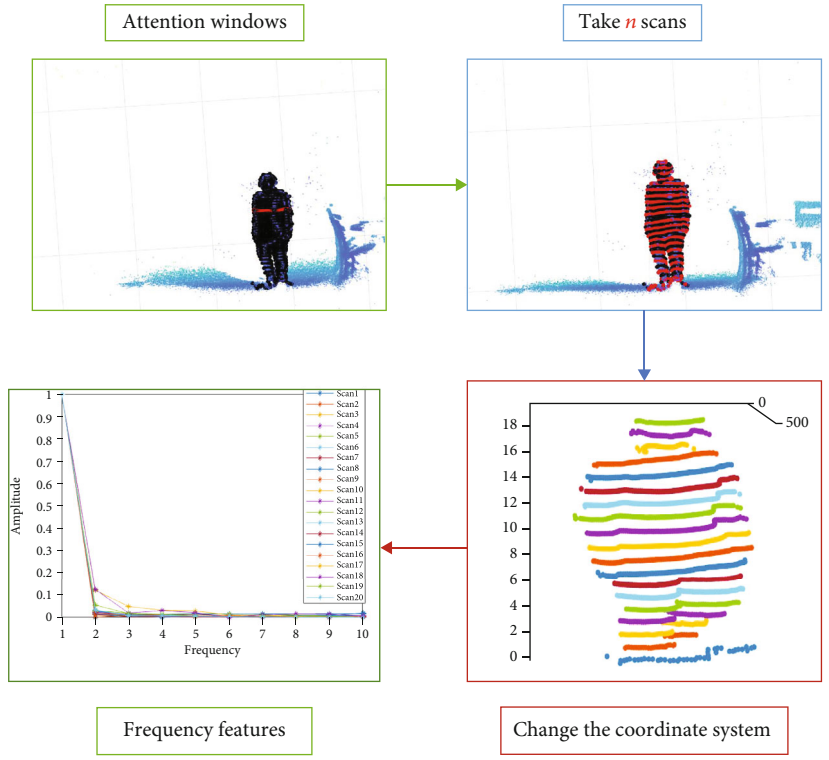

Figure 5: Description of frequency feature extraction: (a) shows the extracted object using RoI, (b) select n scans from the detected scans, (c) changing the coordinate system, and (d) amplitude of performed FFT on selected scans.

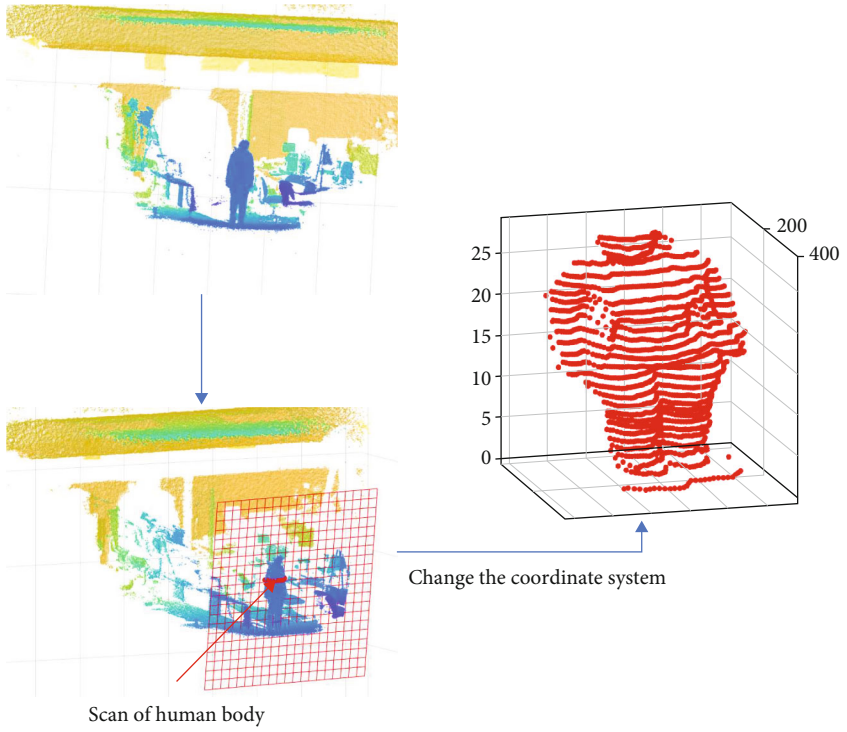

Figure 6: The plane generated from the points of the body and the human body in the new coordinate system.

belongs to the subject s in frame f. In each frame, N_{sf} can be different depending on the size of the subject and its position with respect to the depth sensor. For the purpose of posture detection, n number of these scans are utilized where $n < N_{sf}$. We will analyze and discuss the contribution of n to our performance evaluation results. Having a constant number

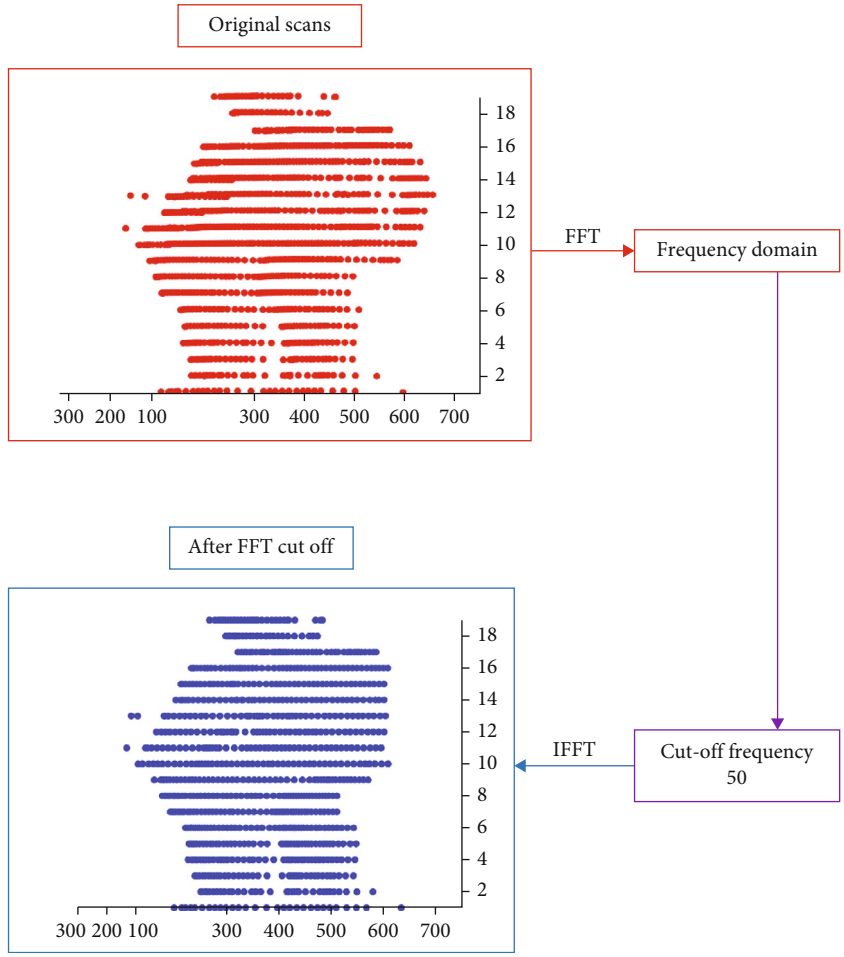

Figure 7: The effect of FFT on the scans (a) and the original scans (b). The scans after converting back from the frequency domain. The cut off frequency is 50.

of scans for all subjects in all of the tracking frames helps to keep the constant size feature vector during the training and testing phases at different locations.

Before extracting the frequency features, we first map the human body to a new coordinate system in which the origin is located on the body coordinate frame, i.e., the origin of the body coordinate frame system and its orientation is defined in the plane passing through the human body. The rotation matrix for mapping the sensor coordinate system to the new coordinate system (passing through the human body) can be calculated using principal component analysis (PCA) of the points belonging to the subject, and the fact that one of the axes of the coordinate of the frame is perpendicular to the calculated plane. The origin of the coordinate system will be located at $O = \{\text{med}(x_{ij}), \text{med}(y_{ij}), \min(z_{ij})\}$. The new coordinate system of the example in Figure 5(a)) is shown in Figure 5(c)). Figure 6 illustrates the plane and the result of the change in the coordinate system.

One of the important advantages of frequency feature extraction is that the noise can be filtered easily by cutting off the higher frequency components. Figure 7 illustrates the effect of this property using an example. Figure 7(a)) shows the original scans. After taking the FFT and removing the higher frequencies (cut off frequency is $f = 50$) and converting it back through inverse FFT, the scans are smoother and hence contain only the main information regarding the orientation or posture of the subject. Besides, the feature number for each scan remains similar and independent of the number of points in each scan.

Let S_{ai} where $ai = [1 \cdots n]$ be one of the selected scans from the subject point cloud. We up sample each scan to $I \geq \max(\forall_{i=1}^{n} size(s_{ai}))$ where I is constant for all the scans of all the subjects in all of the frames. This step is necessary to ensure that the extracted frequencies contain the signal information, and the features are similar in all samples of scans.

In the next step, we perform the FFT of each signal after filtering out the higher frequency components and keeping only m first dominant frequencies. Finally, the selected frequencies in a given 1D scan are also employed as a part of the enhanced features for the classification.

4.2. Classification Algorithms. In this subsection, we present details of the classification methods based on the extracted frequency features. The importance of classification based on frequency features in comparison with the spatial domain features is that it is not dependent on the amplitude of the signal. This offers two advantages: (a) it is less dependent

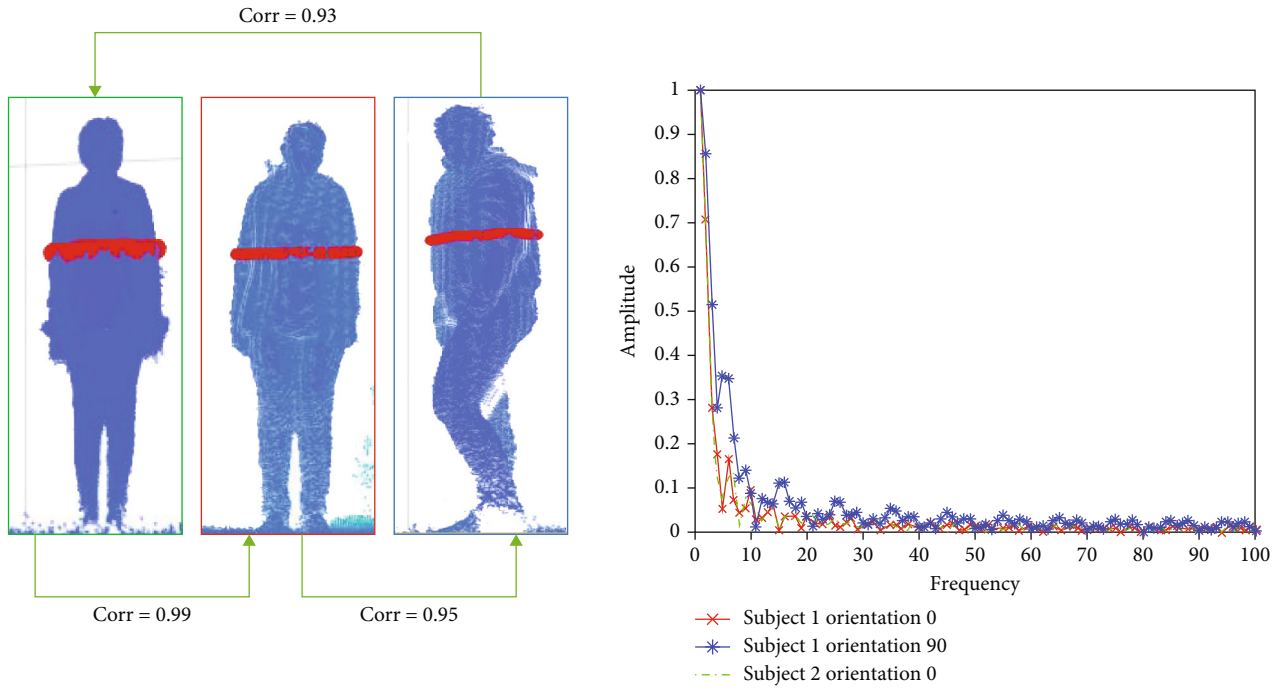

Figure 8: The correlation between the amplitude of the frequency of a scan of two subjects in same orientation and same subject in different orientation. The graph shows each of the scan's frequency, and the value of the correlation is shown between pair of the images.

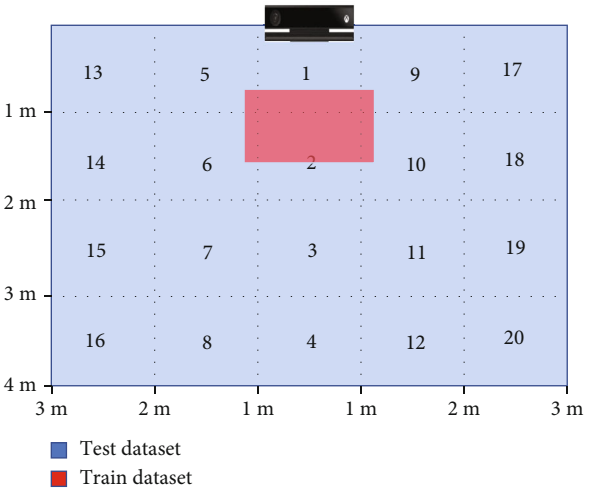

Figure 9: The area undertest the blue area is utilized for testing and the red area is utilized for training.

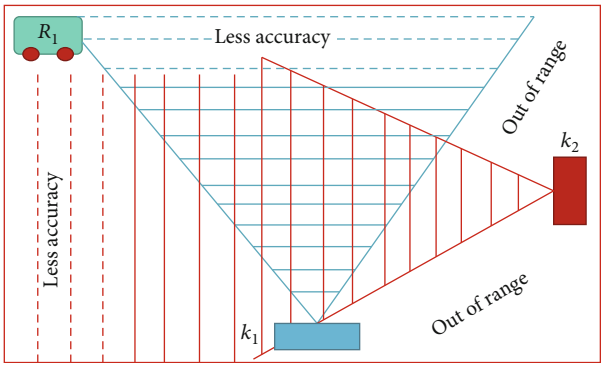

Figure 10: An example of stationary sensors and robot sensor setup we have utilized in this study. The process of selecting sensor modes and role of Schedualer.

on the body shape of the subject, and (b) it is independent of the position of the subject with respect to the sensor. Figure 8 shows an example of frequency features of two subjects in the same postures. The first image belongs to a woman at age 25 while the second and third ones belong to a woman in her 30 with a higher mass body. The corresponding scans are shown by red points in each image, and the correlations between the frequencies of the scans are also shown in the image. The correlation between the frequencies suggests a high similarity between the scans taken from different subjects in the same posture. These two features are explored in order to reduce the number of training sets which is one important step for minimalization purposes.

The training set is obtained at a single location in the monitoring area. For the training data set, a single subject is asked to stand in front of a sensor at a distance of 1 m from the sensor plane. The test dataset is then collected from various subjects at different locations with respect to the sensors. Figure 9 illustrates the locations where the test images are captured (blue areas) and also shows the location of the single training dataset (red area).

For both training and testing datasets and each subject, we perform n scans and for each of the scans (as performed above), we computed the first m frequency profiles that are utilized as the features. The effect of setting different values for m and n is also analyzed and is presented in the next

Table 1: Comparison between two sensors type used in our experimental setup name Microsoft Kinect V2 (stationary sensors (k_1 and k_2)) and Astra Orbbec (robot sensor).

Sensor	k_1 Stationary	k_2 Stationary	R_1 Robot
Type	Kinect V2	Kinect V2	Orbbec Astra
Technology	ToF	ToF	Structured light
Height	120 cm	120 cm	30 cm
Distance range	0.5-8 (effective 4)	0.5-8 (effective 4)	0.6-8 m
Resolution	512×424	512×424	640×480

section. As a result, for each subject in a frame, a total of $n \times m$ features are extracted. These features are used for classification purposes to detect details associated with the movement of the body. We use the extracted features as $[1, n \times m]$ array as an input to a k-nearest neighbour classification algorithm. The distance between each sample is calculated by Manhattan distance with $k = 3$. For testing, the subjects are asked to freely move and stand in different places toward both stationary sensors and the robot. The result of each sensor detection is presented in Section 6.

5. Sensor Network Scheduler

Using multiple sensors can assist in increasing the field of view (FoV) of the monitoring area and can also improve the accuracy of the tracking. In addition, when sensing is also distributed between stationary and mobile sensing platforms, it allows the creation of an adaptive monitoring environment where the overall field of view can be adjusted and reconfigured. However, taking multiple images from different sensors will not necessarily improve the accuracy of the tracking. The movement of the target can trigger more than one sensor, and hence a scheduler is needed to manage the sensors and select the most qualified stationary sensors or reconfigure the position of the mobile sensors. In this section, we present our method for scheduling multiple sensors. Let f_{n_1} be the frame when the subject enters the field of view of one of the stationary sensors. Let the orientation of the subject with respect to the sensor to be $\alpha = \{0, 90\}$ (one of two main orientations shown in Figure 4) with its position with respect to the sensor frame f_1 be defined as $\{x_{f1}, y_{f1}, z_{f1}\}$, and the subject's position in the frame f_{m_2} is $\{x_{fm}, y_{fm}, z_{fm}\}$. The sense of direction of movement of the subject with respect to the sensor can be classified as (a) movement toward a stationary sensor, (b) movement toward the other stationary sensor or the current position of a mobile robot sensor, and (c) movement away from the current coverage of the stationary sensors. Given the above scenarios, the scheduler should perform two tasks: first, it should assign the most qualified sensor for the posture estimation and second, it should plan to navigate the mobile robot sensor toward a direction that can have the subject in its FoV when moving away from the current coverage of the stationary sensors. In the following, first, we present the algorithm which defines the most qualified sensor which can be used for tracking, and then we define the mobile sensor navigation algorithm which enables us to keep the subject in the FoV of the mobile sensor and hence extending the coverage area.

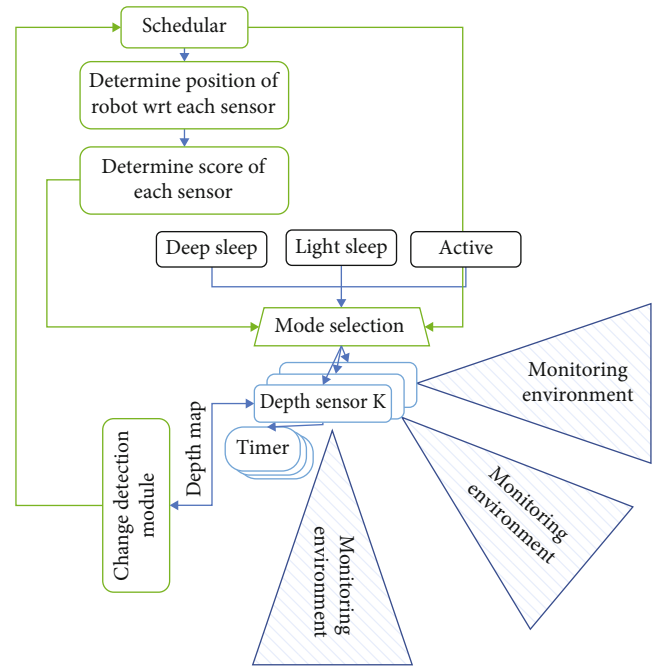

Figure 11: The process of selecting sensor modes and role of scheduler.

5.1. Sensor Rating for Tracking. Figure 10 illustrates an example of an instance of multisensor scenario along with a mobile sensor with each sensor FoV and accuracy of posture estimation. The objective of the design of the sensor scheduler is to better manage the accuracy in estimating details associated with the posture of the subject (such as the position with respect to sensors and the state of movement). Two main factors can affect the accuracy of the estimation: (1) the position of the target with respect to the sensor and (2) the accuracy of the classifier. As the subject gets closer to the sensor, it results in a higher resolution of observation of the sensed information. As a result, it can lead to a more accurate posture estimation and some specific orientations of the subject (as will be discussed in Section 6.3). Table 1 shows the features of stationary and the robot sensor.

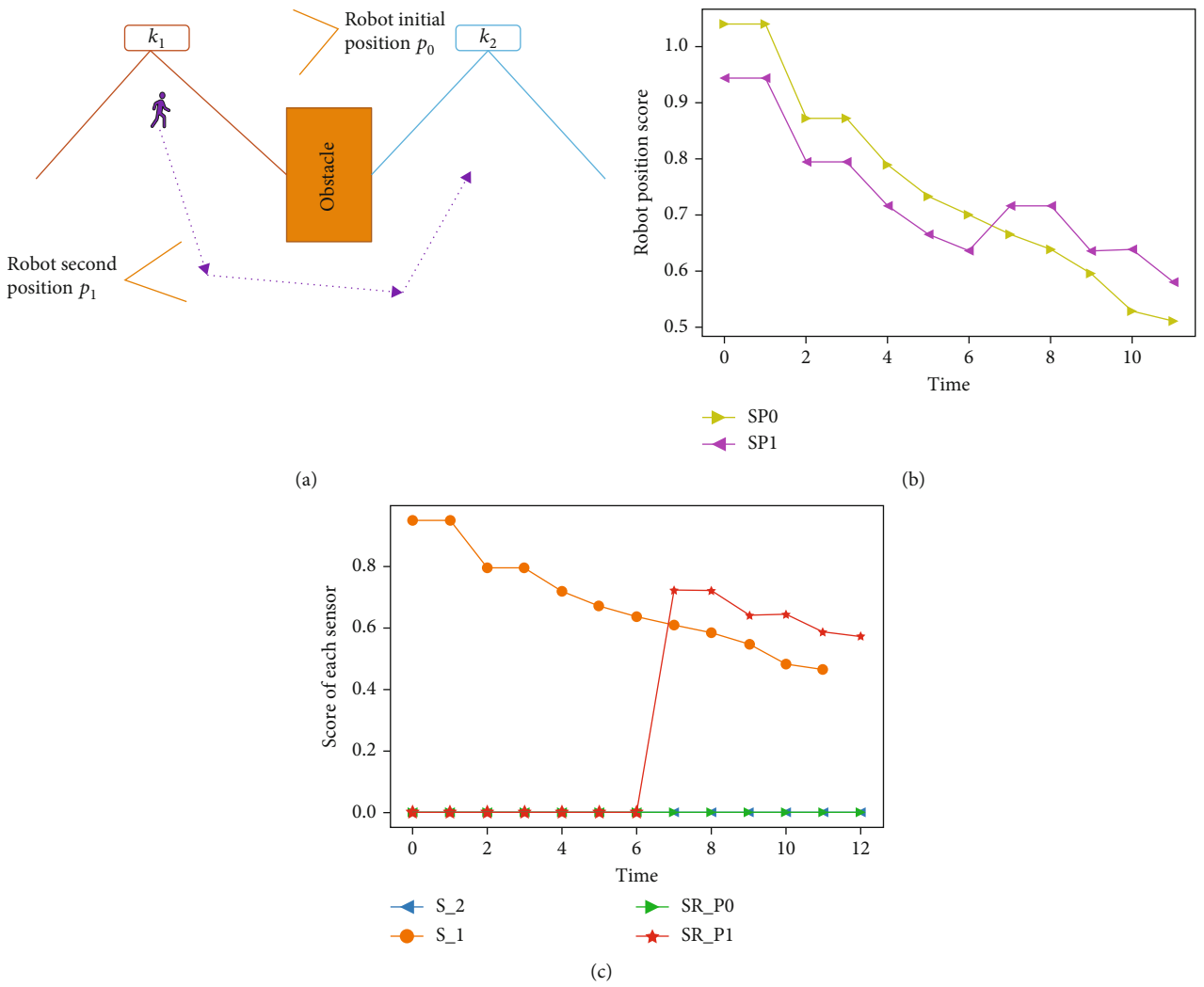

FIGURE 12: (a) The location of stationary sensors (K_1 and k_2) and two predefined location of the robot (p_0 and p_1). The trajectory of the subject's movement is shown by purple dashed lines. (b) Score of each of the sensors (S_1 and S_2 and the robot in position 0 and position 1 SR_{p0} and SR_{p1}). Two predefined position is determined for the robot, where SR_{p0} and SR_{p1} are the score of the robot sensor on any of theses positions. (c) The score of each of the position of the robot calculated by equation (2).

Sensors constantly perform scans to detect any changes in the environment for every $T = t_1$ where T is the period of capturing. If a sensor has detected no changes in the k – th frame, it enters the "deep-sleep" mode for a period of $T = t_2$ where $t_2 > t_1$. On the other hand, if a change occurs in the field of view of any of the sensors, it then computes the distance of the subject to the sensor. The sensor that is the closest to the target is assigned to the task of further detecting the subject and its posture estimation while other sensors enter the "light-sleep" in which $T = t_3$ where $t_1 < t_3 < t_2$. Let the distance of the subject with respect to a sensor be measured as d (in meter) and let its average accuracy in a test dataset for detection at distance d be given as A%. For a given orientation of the target with respect to a sensor, we can now define a tracking score as

$$S = a_1 A + \frac{a_2}{d} + a_3 \theta, \tag{1}$$

where $s \leq 1$ is the scor,e and a_1, a_2, and a_3 are the normalization factors so that the optimized condition provides $S = 1$. Sensors which can detect the target will calculate their distance, percentage accuracy, and relative orientation toward the subject, and the one with a closer distance takes over the tracking task. This approach can provide an advantage for continuous tracking while avoiding unnecessary switching between sensors. The scheduler reduces energy consumption while keeping the accuracy of the posture estimation accurate. Figure 11 shows the process of selecting the mode for each stationary sensor and the role of scheduler with an example.

5.2. Mobile Sensor Selection. A sensor which is mounted on a mobile platform (e.g., robot) is utilized in order to extend the monitoring area. It can provide sensed information about the subject when the stationary sensors are not able to. It can be used to increase the accuracy of detection of the target in the FoV of stationary sensors in the presence of a low

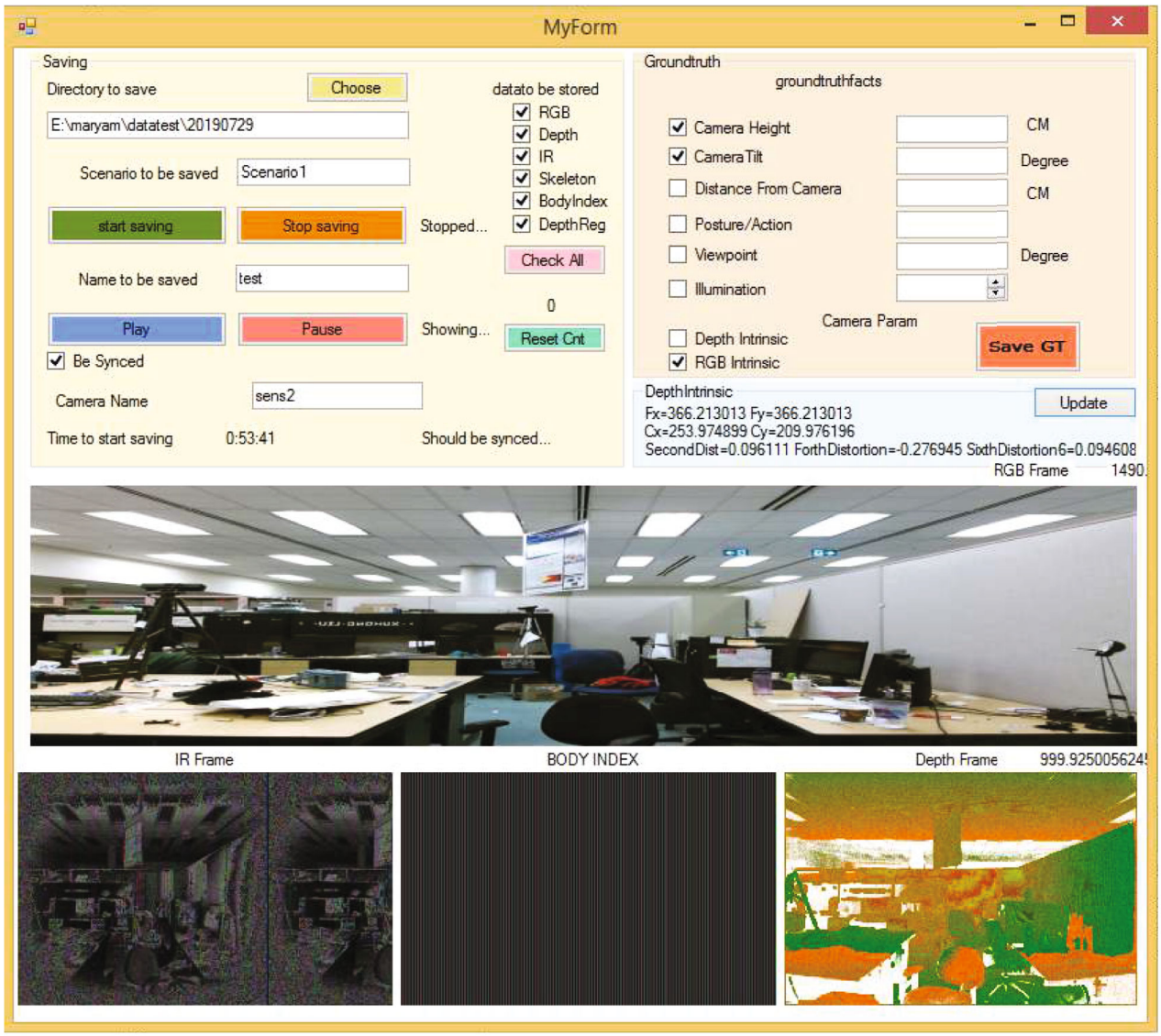

Figure 13: The designed GUI related to the custom API design in order to collect the required dataset.

tracking score (i.e., the low value in S as introduced in equation (1)). In these events, the mobile robot sensor can move to a location that can offer a higher value of tracking score for better monitoring of the subject. As such, the predicted direction of movement of the subject defined by the stationary sensors can be utilized to plan the trajectory of the mobile sensor in the monitoring area (to be presented in the next section). Figure 12 illustrates an example of the robot trajectory and the performance of the scheduler to track the subject. Here, we assume some predefined locations in the monitoring area to navigate the mobile robot sensor to those locations. These locations are selected based on the estimation of the next position of the subject where the robot will be navigated to one of these predefined locations. Besides, the robot's navigation to new locations is programmed to minimize the attention to the robot sensor (i.e., the path passes closer to the walls). The same algorithm of change detection has been used by the mobile sensor to detect the movement of the subject when entering its field of view. Using the mobile sensor is beneficial especially to minimize the number of stationary sensors by avoiding putting any sensors in less crowded areas or less used areas.

5.3. Navigation of the Mobile Sensor. To navigate the robotic sensors to the locations which are within the range of the subject, the schedular uses the position of the subject with respect to each of the stationary sensors. The goal is to keep the S (in equation (1)) as maximum as possible (close to one). Meanwhile, the robot should avoid unnecessary movements in order to preserve energy and to be inconspicuous and out of the way of the subject. Some predefined locations are determined for the robot sensor in order to cover the out of the field of view areas of the stationary sensors. As the subject moves toward any of these positions, the schedular would predict the best location for the robot to be navigated to.

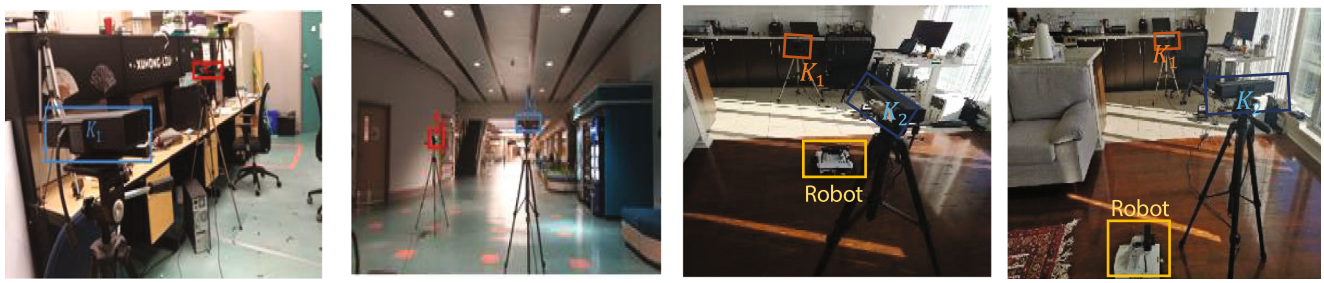

Figure 14: Various experimental setups used in this paper show the position of the stationary camera in the laboratory setup and a hall-way combined with the stationary cameras and the mobile one. The later experiments were carried in the actual living space.

Table 2: The accuracy of the proposed change detection algorithm in different distances from the camera and its comparison with the method proposed in [54].

D (m form sensor plane)	Sensor type	Noise characteristic method (proposed method)	Sensor characteristic method ([54])
4	Kinect	96.65	88.58
4	Astra Orbbec	95.54	80.23
3.5	Kinect	97.24	94.35
3.5	Astra Orbbec	95.32	87.5
3	Kinect	97.24	95.6
3	Astra Orbbec	96.2	88.3
2.5	Kinect	98.84	95.6
2.5	Astra Orbbec	98	90.32
≤2	Kinect	98.84	95.6
≤2	Astra Orbbec	98	94.52

Assume we have defined M predefined locations for the robot, our problem is to find the new position of the robot as p_{prob} which we can be formulated as follows:

$$\text{new}_{\text{pos}} = \arg_{\max}\left(\max_{0\le l\le M} s_{\text{p}_{\text{rob}_l}}\right),$$
$$s_{\text{p}_{\text{rob}}} = \varsigma_{\text{p}_{\text{rob}}}\left(\max\left(\max_{K_n:1\le n<N} S_{k_n}\left(P_{\text{suj}_{\text{pre}}}\right), S_{\text{rob}}\left(P_{\text{suj}_{\text{pre}}}\right)\right)\right), \quad (2)$$

where new_{pos} is the next position that the robot should move to, s_{k_n} is the score of the stationary sensor k_n, and S_{rob} is the score of the robot sensor (calculated in equation (1)). $p_{\text{suj}_{\text{pre}}}$ is the predicted position of the subject and $s_{\text{P}_{\text{robot}}}$ is the score of each robot position. Equation (2) determines the next location of the robot among predefined locations. The decision is based on the ranking which maximizes the overall score of the sensors (as defined in equation one). In this equation $\varsigma = 1.1$, if the $p_{\text{rob}} = p_{\text{rob}_{\text{current}}}$ (calculate the value of f for the current position of the robot), it is equal to one; otherwise, $p_{\text{suj}_{\text{pre}}}$ is the prediction of the new position of the target.

To find out the position of the robot, a rough estimation of the subject is needed. To achieve this, Kalman filter has been utilized to predict the next position of the subject. The estimated parameters are x and y of the subject based on their current and previous positions. Figure 12(b) illustrates the score of each of the sensors (S_1 and S_2 and the robot in position 0 and position 1 SR_{p0} and SR_{p1}) from the scenario depicted in Figure 12(a). Figure 12(c) shows the score of each of the positions of the robot calculated by equation (2).

It is worth emphasizing that in our proposed method, we are not fusing the data; instead, we are using the scoring methodology to select the most accurate sensor. In fact, the purpose of using multiple sensor is to expand the field of view of the monitoring system.

6. Experimental Study and Evaluation

In this section, we study the experimental results following the methods and methodologies presented in the previous section. We will first explain the data which has been utilized in this research followed by their evaluation.

6.1. Dataset Collection. There exists a large number of datasets containing various human postures and actions. However, the scope of this paper does not fit within any of those, and hence we collect our dataset. The reason is that we are looking for a piece of very detailed information about human posture. While often the published datasets are used to classify images based on the higher level of information they contain (e.g. as related to action classification), it makes it hard to filter out the information that is needed for this study. Therefore, we collected our dataset. To do so, we design and implement an API to collect the dataset (it will be made public in the near future). Figure 13 shows the screenshot of the API. It is implemented in a modular way so that it can be exchanged with any other sensors rather than Kinect V2. In addition, the API can synchronize multiple sensors using universal clock of the internet in a distributed matter. The collected dataset contains 5 subjects from age 25 to 35 and 3 women and 2 men with different body shapes. Figure 14 shows the setup of sensors and the location where the experiment is carried out. The experiment is collected in various locations including a clutter lab where the sensors' FOV was restricted by various objects or a hallway where it is captured out of range of the sensor.

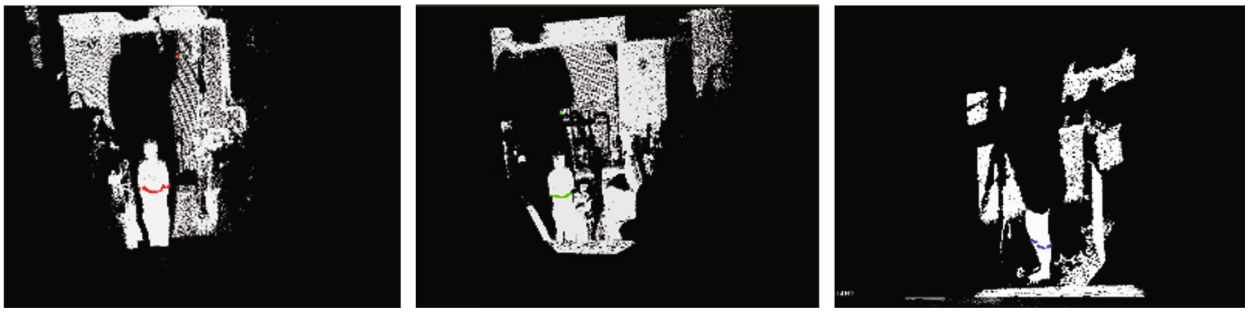

Figure 15: The point cloud captured by K_1 and K_2 and robot (R) where the changes are shown by red, green, and blue points, respectively.

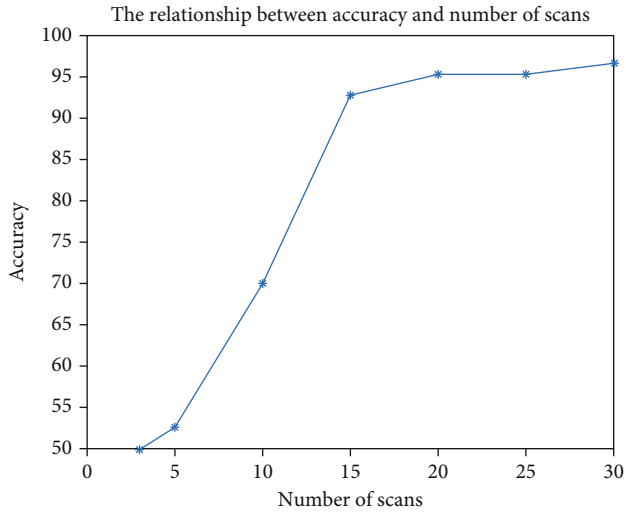

Figure 16: The accuracy of the classification vs. the number of scans. As the number of scans increases, the accuracy increases. However, after 15 scans, the accuracy curve levels.

6.2. Change Detection. The proposed change detection algorithm used the characteristics of the noise to differentiate between the change and false positive (noise). We are comparing the proposed method with [54] that uses the characteristic of the time of flight sensor to differentiate between noise and change detection due to real changes in the scene. The results of the comparison between the two methods are shown in Table 2. Two types of sensors are utilized for the comparison. The first sensor is a Microsoft Kinect V2 which uses time of flight technology for capturing the depth, and the second sensor is the Orbbec Astra depth sensor which uses structured light technology for capturing depth data (the detailed feature of each sensor is shown in Table 1). The example of detected change by any of the sensors used in this experiment (two stationary sensors (K_1 and K_2) and a robot sensor R) is shown in Figure 15. D is the distance between the subject and the sensor plane. The results show the change detection when the scene captured by the Orbbec sensor is improved distinctly while the change detection in the depth map captured by Kinect sensor is slightly better than the method in [54]. The metric that is shown in the tables are rec = TP/TP + FN .

Table 3: The accuracy of orientation detection in different scenarios where the training set is capture data 1 m from the sensor plane and the test is from a different subject in distance 1 m in scenario 1, and the captured test set is in a distance of 1 m, 2 m, and 3 m from sensors' plane in Sc2, Sc3, and Sc4, respectively.

Number of FFT profile	Model	Sc 1	Sc 2	Sc 3	Sc 4
FFT profile = 5	KNN	51.13	62.61	52.94	63.24
	CNN	64.31	71.30	62.75	62.50
FFT profile = 10	KNN	53.38	61.74	64.71	83.09
	CNN	6817	80.87	79.74	58.09
FFT profile = 15	KNN	7138	95.65	90.20	46.32
	CNN	72.99	77.39	66.67	38.24
FFT profile = 20	KNN	80.39	94.78	70.59	38.24
	CNN	7846	84.35	60.78	38.24
FFT profile = 25	KNN	85.53	92.17	56.86	38.24
	CNN	81.35	85.22	56.21	38.24
FFT profile = 30	KNN	86.82	88.70	56.21	38.24
	CNN	82.32	84.35	52.94	38.24
FFT profile = 35	KNN	85.85	86.09	56.86	38.24
	CNN	83.92	82.61	49.67	38.24
FFT profile = 40	KNN	85.53	8783	56.86	38.24
	CNN	83.60	74.78	49.02	38.24
FFT profile = 45	KNN	84.24	87.83	56.86	38.24
	CNN	81.99	73.91	49.67	38.24
FFT profile = 50	KNN	83.28	88.70	58.17	38.24
	CNN	81.99	76.52	52.29	38.24
FFT profile = 100	KNN	83.92	90.43	56.86	38.24
	CNN	81.99	79.13	52.94	38.24
FFT profile = 150	KNN	83.92	89.57	58.17	38.24
	CNN	81.99	81.74	53.59	38.24

6.3. Parametric Analysis and Evaluation. As we discuss in Section 4, the size of the feature vector is $n \times m$ where n is the number of scans, and m is the number of frequencies (cut off frequency). Without upsampling of the scans, we analyze the influence of these two factors (namely, n and m) on classifying the orientation of the subject with respect to the sensor. Figure 16 shows the effect of the number of scans on the result. In this scenario, the value of m kept constant. As can be seen in the table, the higher the number of

Table 4: The final accuracy of estimating the orientation of the subject after adding upsampling layer in different scenarios where the training set is captured in 1 m from the sensor and the test is from a totally different subject in distance 1 m in scenario 1, and the captured test set is in a distance of 1 m, 2 m, and 3 m in Sc2, Sc3, and Sc4, respectively.

Details	Sc1	Sc 2	S3	Sc 4
FFT profile = 25 Scans = 25	93.65	95.65	96.73	86.76

Table 5: The final accuracy of estimating the posture of the subject in different scenarios where the training set is captured in 1 m from the sensor and the test is 2 m, 3 m, and 4 m in Sc1, Sc2, and Sc3, respectively.

	Sc1	Sc2	sc3
Orientation 0 FFT profile = 25 Scans = 25	78.28	75.52	56.4
Orientation 90 FFT profile = 25 Scans = 25	79.09	72.16	48.44

scans leading to higher accuracy. However, after increasing the number of scans to more than 20, the result is not improved significantly. The reason is that the extra scans are mostly adding redundant information which cannot improve the model drastically. In this figure (Figure 16), the test and train datasets both are collected from one subject in the validation.

The other element in the proposed approach is the cut off frequency. Table 3 shows the result of the classification accuracy with various cutoff frequencies. We also have defined four different scenarios:

(1) Scenario 1: both training and test set are from 1 m from the sensor plane (grid 1 in Figure 9) while the subject is different in each set

(2) Scenario 2: the training set captured 1 m from the sensor plane while the test set is captured at a distance of 2 m from the sensor plane (grid numbers 2, 5, and 6 in Figure 9)

(3) Scenario 3: the training set captured 1 m from the sensor plane while the test set is captured at the distance of 3 m from the sensor plane (grid numbers 3, 6, 9, 11, and 13 in Figure 9)

(4) Scenario 4: the training set captured 1 m from the sensor plane while the test set is captured at the distance of 4 m from the sensor plane (grid numbers 4, 7, 10, 12, and 14 in Figure 9). To provide a comparison with deep learning approaches, a convolutional neural network with 3 fully connected convolutional layers and 1 fully connected layers has been used for classification purposes. The input of the network is a matrix of the n scans and m FFT profiles. As we expected, the CNN is less accurate than the KNN as they usually need the high amount of data

As can be seen, the higher value of m results in lower accuracy since it contains irrelevant information and noise. The lower cut-off frequency has a better result especially when farther the distances are. In addition, the higher value of m causes overfitting of the models. This issue can be avoided by increasing the number of images in the training set.

Table 4 shows the results when an extra step of upsampling is added. This results in achieving higher accuracy in estimating the orientation of the subject.

To classify the posture of the subjects in one of two defined classes as walking (where one of the legs is in front of the other) and standing (where two legs are next to each other), we utilized the KNN model as well. Here, we have defined three scenarios as well:

(1) Scenario 1: the training set is captured in 1 m from the sensor plane (grid 1 in Figure 6) while the test set is captured at the distance of 2 m from the sensor plane (grid numbers 2, 5, and 6 in Figure 6)

(2) Scenario 2: the training set is captured at 1 m from the sensor plane while the test set is captured at the distance of 3 m from the sensor plane (grid numbers 3, 6, 9, 11, and 13 in Figure 6)

(3) Scenario 3: the training set is captured at 1 m from the sensor plane while the test set is captured at the distance of 4 m from the sensor plane (grid numbers 4, 7, 10, 12, and 14 in Figure 6)

Table 5 shows the final result of estimating two defined postures.

6.4. Multisensor Evaluation and Scheduling. In this section, we have defined two different scenarios to evaluate the performance of the scheduler which are shown in Figures 17 and 18. For this study, we have an observation room with a dimension is 5 by 3 meters. Figures 17(a) and 18(a) show the trajectory of the subject in the field of view in each scenario shown in purple dashed lines. Each of these two scenarios was repeated 3 times, and the average score values are shown in Figures 17(c) and 18(c), respectively. In the first scenario shown in Figure 17(a), the monitoring area is covered by two stationary sensors with some overlap; however, parts of the monitoring environment remain uncovered. The stationary sensors are 4 meters apart, and the robot sensor is located 5 meters from the first sensor. Two positions are defined for the robot sensor to complete the monitoring coverage. The subject started from sensor K_1 field of view and move toward the uncovered area as the subject moves outside the field of view of the sensor, and the schedular navigates the robot sensor to the next position. Figure 17(c) shows the scores of each of the sensors and the robot. In this scenario, as the subject moves outside of the sensor's FOV, the score of each sensor will be reduced until the robot is relocated by the scheduler to a new position with a better overview of the subject with an improved score. When the subject enters a visible area of the stationary cameras, the robot will then discontinue its tracking and goes to

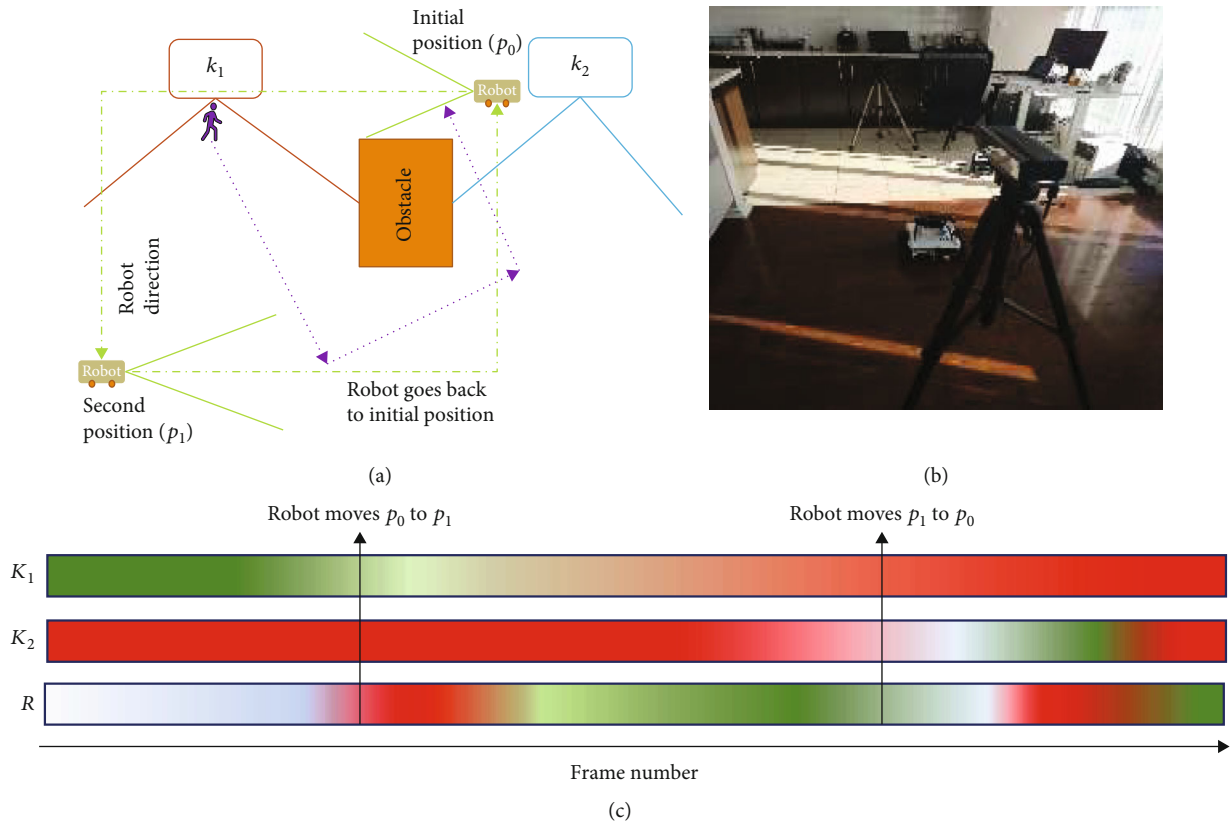

Figure 17: (a) The setup with trajectory of the subject moving in the monitoring area. (b) The image of actual setup. (c) The representation of sensor's situation in cooperating for target tracking, the shaded green shows that the target is tracked by the sensor. Blue is the presentation of detection, and the red showed no detection of the target.

its initial location. In the second scenario shown in Figure 18, the subject enters the field of view of the second sensor and moves toward the robot when the subject is not discovered by any of the sensors anymore, and their score is reduced shown in Figure 18(c). The robot moves to the second predefined position based on the estimation of the previous subject movement and continue to monitor the subject. The scores that are shown in Figures 17(c) and 18(c) are the value calculated by equation (1) separately by each sensor, and the color is assigned to each number from zero as red and one as green.

6.5. Real Time Implementation. The schedular and the whole setup is implemented real-time using ROS framework. The setup includes two Microsoft Kinect V2 and one mobile robot which includes one Orbbec depth sensor. The communication of stationary depth sensors and a mobile robot is established using ROS framework. The details of the communications and message passing of the sensors are shown in Figure 19. The time complexity for the change detection algorithm can be calculated by considering that each scan is taken from one place, and W is the width of the depth map. FFT complexity analysis results in O(wlg2w) in scanning. The complexity for change detection using SVM model and pose detection algorithm for n scans would be O(nwlg2w). The scheduler complexity depends on the number of stationary sensors (K), robot sensors (R), and the position of robot sensor (P). The scheduler should find the most qualified sensors at $O(K + R)$ in the worst case scenario and determine the best position for the robot sensor with O(P). The use of Kalman filter for forecasting each robot sensor that has only 2 parameters will result in the complexity of the robot sensors to be $O(R)$ which should be determined for each position, and hence the complexity of the scheduler would be $O(K + R + PR)$. Table 6 shows the list of hardware and software components used in this study where Table 7 shows the comparison of depth sensors used in the study regarding their range, resolution, frame rate, and FOV.

6.6. Summary of Evaluation. In this section, we have evaluated our proposed method in various scenarios. First, we compared our change detection algorithm with that of [54] in Table 2. Then, we showed how the different parameters are impacting the proposed algorithm in Table 3 as the result and by selecting the best parameters, we showed the performance of the proposed posture detection in detailed posture estimation and orientation toward the sensor in Tables 4 and 5, respectively. Then, we evaluate our proposed scheduler in two separate scenarios with two stationary sensors and 1 mobile sensor. Finally, we showed the flow of real time implementation of the proposed work.

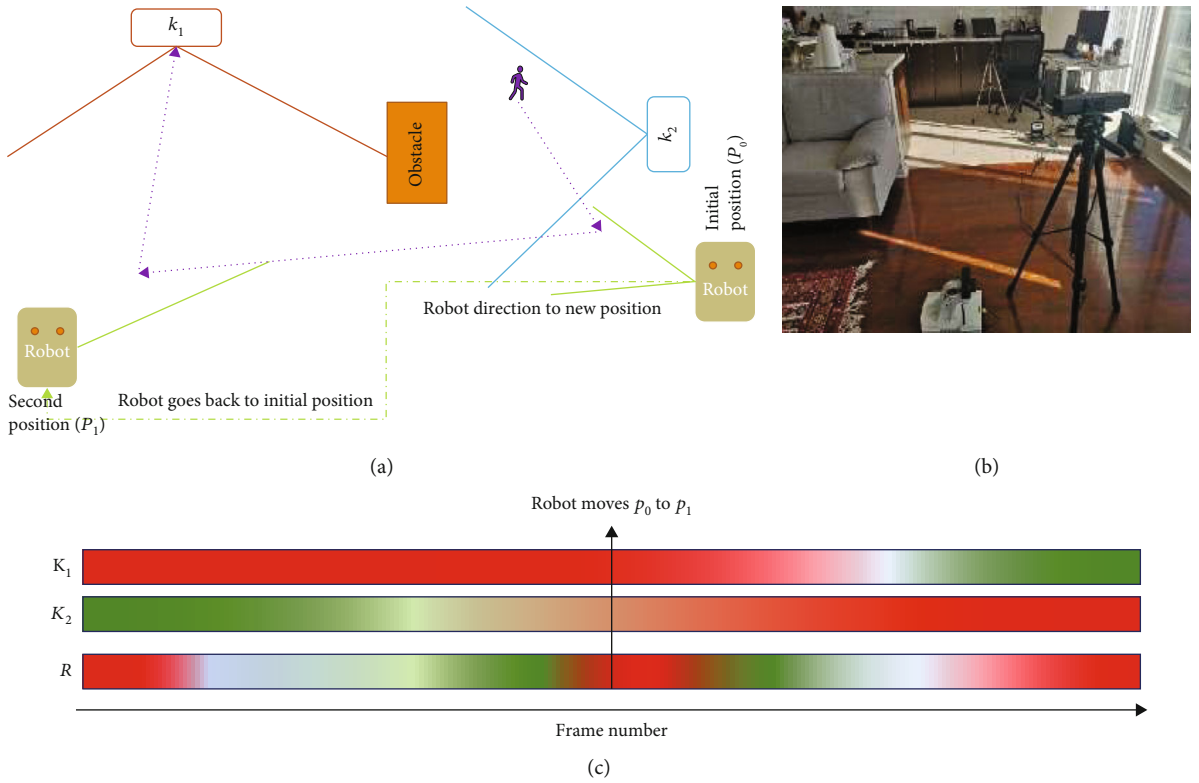

Figure 18: (a) The setup with trajectory of the subject moving in the monitoring area. (b) The image of actual setup. (c) The representation of sensor's situation in cooperating for target tracking, the shaded green shows that the target is tracked by the sensor. Blue is the presentation of detection, and the red showed no detection of the target.

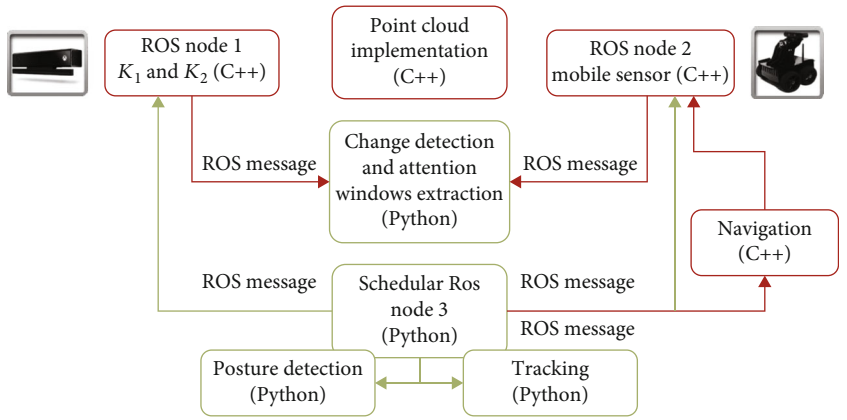

Figure 19: The details of network sensor communication.

Table 6: The equipment, software, frameworks, and programing languages used in this study.

Hardware	Kinect V2.0 (stationary sensor) × 2		Orbbec Astra (robot sensor) × 1
Software and frameworks and programming language	ROS	Open Kinect, open NI	Python, C++, C#.net

Table 7: Comparison of sensors used in the study.

Name	Range (m)	Resolution	Framerate	FOV (degrees)
Kinect V2	8 (reliable till 4.5)	512 × 424	30	89 × 71
Orbbec Astra	0.6–8	640 × 480	30	60 × 49.5

7. Conclusions and Future Work

In this study, we proposed a novel and efficient method to minimize the training data that can be utilized for data minimization in a network of depth sensors. We use the notation of 1D scans and their frequency features to detect the changes in the sensors' FoV and estimate the posture and position of the subject with respect to the sensor. Using 1D scans offers two main advantages. Firstly, they reduce the necessary pixels of the depth map and hence improve the computation complexity and increase the time efficiency of the system in a network of stationery and robot sensors. Secondly, they enable to efficient use the frequency features of the depth maps to decrease the sensitivity of variant distance between the sensor and the subject which leads to a reduction in the number of training data.

A dynamic RoI model based on a novel change detection method has been designed on the top layer of 1D scans in order to detect the changes in the scene and separate the background from the foreground. In [59, 60], we showed that using 1D scans is beneficial over skeleton point extraction methods which are generally state of the art in pose detection. In addition, we have shown that using only 20 1D scans provides a good estimation of the posture.

Finally, we proposed a scheduler to handle multiple stationery and robot sensors to save energy and increase the accuracy of the monitoring. Our proposed network contains two stationary ToF sensors (Kinect V2) and one robot structural light sensor (Astra Orbbec). The schedular is responsible to assign the most qualified sensors for the tracking and estimation task based on the scoring schematic. A score of each sensor is calculated based on the accuracy of the posture estimation and the location of the sensor with respect to each sensor. In addition, the schedular navigates a robot sensor to extend the monitoring area and improve the accuracy of posture estimation. Some predefined locations are determined for the robot sensor and based on the estimation of the score of the whole system, one of those locations will be selected, and the robot is navigated to the new location if needed. A Kalman filter has been utilized to improve robot navigation by estimating the future location of the subject in the monitoring area. It should be noted that in the current setup, the robot is used to cover the less crowded areas of the monitoring environment. If the subject frequently moves to the uncovered areas, the criterion for the energy efficiency proposed in this work would not be satisfied. In addition, we assumed that the moving trajectory of the robot is a cluttered free environment. Otherwise, various available collision detection algorithms can be utilized for enhancing the planned trajectory.

In future work, the robot can benefit from various navigation methods to follow the subject directly, e.g., [61]. More postures and subject orientations toward sensors can be defined using the same proposed method, and also the data set can be enhanced using more various subjects. Furthermore, the extended method can be utilized to estimate and track multiple subjects occupying more comprehensive postures and locations in the monitoring area. The RGB camera also can be used to enhance the accuracy of the estimation; however, it might increase the computational complexity of the proposed method.

In addition to depth and RGB monitoring systems, using a wearable sensor is another alternative to detect and analyze the human posture; among them, methods such as [62–69] can be mentioned. Although the accuracy of these methods is very promising, they need the voluntary cooperation of the subject, and they are prone to be forgotten. A comparison of these methods can provide a benchmark for the accuracy that is possible to achieve by vision-based methods such as the depth sensor proposed in this paper.

Nomenclature

s_a: 1D scan from depth profile
S: Overall score of each sensor in monitoring system
p_{rob}: Position of the robot
$s_{p_{\text{rob}}}$: Score of the position of the robot
$p_{\text{suj}_{\text{pre}}}$: Probability of subject position.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded through support from the Natural Sciences and Engineering Research Council of Canada.

References

[1] Z. Chen, R. Wu, Y. Lin et al., "Plant disease recognition model based on improved YOLOv5," *Agronomy*, vol. 12, no. 2, p. 365, 2022.

[2] Y. Tang, M. Chen, Y. Lin, X. Huang, Y. He, and L. Li, "Vision-based three-dimensional reconstruction and monitoring of large-scale steel tubular structures," *Advances in Civil Engineering*, vol. 2020, 1236017 pages, 2020.

[3] F. Wu, J. Duan, S. Chen, Y. Ye, P. Ai, and Z. Yang, "Multi-target recognition of bananas and automatic positioning for the inflorescence axis cutting point," *Frontiers in Plant Science*, vol. 12, article 705021, 2021.

[4] Y. Tang, M. Zhu, Z. Chen et al., "Seismic performance evaluation of recycled aggregate concrete-filled steel tubular columns with field strain detected via a novel mark-free vision method," *Structure*, vol. 37, pp. 426–441, 2022.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, San Diego, CA, USA, June 2005.

[6] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *2006 IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition - Volume 2 (CVPR'06), vol. 2, pp. 1491–1498, New York, NY, USA, June 2006.

[7] T. Watanabe, S. Ito, and K. Yokoi, "Co-occurrence histograms of oriented gradients for pedestrian detection," in *Pacific-Rim Symposium on Image and Video Technology*, pp. 37–47, 2009.

[8] Y. Qi, S. Zhang, L. Qin et al., "Hedged deep tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4303–4311, Las Vegas, NV, USA, June 2016.

[9] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: hard positive generation via adversary for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2606–2615, Honolulu, HI, USA, July 2017.

[10] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, Cham, 2015.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[13] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1925–1934, Honolulu, HI, USA, July 2017.

[14] M. Munaro, F. Basso, and E. Menegatti, "OpenPTrack: open source multi-camera calibration and people tracking for RGB-D camera networks," *Robotics and Autonomous Systems*, vol. 75, pp. 525–538, 2016.

[15] M. Carraro, M. Munaro, and E. Menegatti, "Cost-efficient RGB-D smart camera for people detection and tracking," *Journal of Electronic Imaging*, vol. 25, no. 4, p. 041007, 2016.

[16] M. Carraro, M. Munaro, J. Burke, and E. Menegatti, "Real-time marker-less multi-person 3D pose estimation in RGB-depth camera networks," in *International Conference on Intelligent Autonomous Systems*, pp. 534–545, Springer, Cham, 2019.

[17] K. Koide, E. Menegatti, M. Carraro, M. Munaro, and J. Miura, "People tracking and re-identification by face recognition for RGB-D camera networks," in *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–7, Paris, France, September 2017.

[18] M. Donoser, H. Riemenschneider, and H. Bischof, "Shape guided maximally stable extremal region (MSER) tracking," in *2010 20th International Conference on Pattern Recognition*, pp. 1800–1803, Istanbul, Turkey, August 2010.

[19] V. Parameswaran, V. Ramesh, and I. Zoghlami, "Tunable kernels for tracking," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, vol. 2, pp. 2179–2186, New York, NY, USA, June 2006.

[20] G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, vol. 1, pp. 790–797, Washington, DC, USA, 2004.

[21] M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, June 2007.

[22] A. R. Kosiorek, A. Bewley, and I. Posner, "Hierarchical attentive recurrent tracking," *Nips*, 2017.

[23] C. Blum and D. Merkle, "Swarm intelligence," *Swarm Intell. Optim*, C. Blum and D. Merkle, Eds., pp. 43–85, 2008.

[24] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: do humans and deep networks look at the same regions?," *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[25] A. Roy and S. Todorovic, *A Multi-Scale CNN for Affordance Segmentation in RGB Images*, Springer, Cham, 2016.

[26] J. Chun, S. Park, and M. Ji, "3D human pose estimation from RGB-D images using deep learning method," in *Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing - SSIP 2018*, pp. 51–55, Prague, Czech Republic, 2018.

[27] T. Hu et al., "Improving video segmentation by fusing depth cues and the visual background extractor (ViBe) algorithm," *Syst. Man, Cybern. (SMC), 2017 IEEE Int. Conf.*, vol. 96, no. 5, pp. 76–85, 2017.

[28] T. Ophoff, K. Van Beeck, and T. Goedeme, "Improving real-time pedestrian detectors with RGB+depth fusion," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, Auckland, New Zealand, November 2018.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[30] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5636–5643, Hong Kong, China, May 2014.

[31] T. Linder, S. Wehner, and K. O. Arras, "Real-time full-body human gender recognition in (RGB)-D data," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3039–3045, Seattle, WA, USA, May 2015.

[32] F. Fang, K. Qian, B. Zhou, and X. Ma, "Real-time RGB-D based people detection and tracking system for mobile robots," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1937–1941, Takamatsu, Japan, August 2017.

[33] L. Tian, M. Li, Y. Hao, J. Liu, G. Zhang, and Y. Q. Chen, "Robust 3-d human detection in complex environments with a depth camera," *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2249–2261, 2018.

[34] M. Rasoulidanesh, S. Yadav, S. Herath, Y. Vaghei, and S. Payandeh, "Deep attention models for human tracking using RGBD," *Sensors*, vol. 19, no. 4, p. 750, 2019.

[35] J. Shotton, A. Fitzgibbon, M. Cook et al., "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, Colorado Springs, CO, USA, June 2011.

[36] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1544–1553, Las Vegas, NV, USA, June 2016.

[37] Z. Liu, J. Huang, J. Han, S. Bu, and J. Lv, "Human motion tracking by multiple RGBD cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 2014–2027, 2017.

[38] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB +D: a large scale dataset for 3D human activity analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019, Las Vegas, NV, USA, June 2016.

[39] C. N. Phyo, T. T. Zin, and P. Tin, "Deep learning for recognizing human activities using motions of skeletal joints," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 243–252, 2019.

[40] P. Khaire, P. Kumar, and J. Imran, "Combining CNN streams of RGB-D and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018.

[41] H. Zhang, Y. Ji, W. Huang, and L. Liu, "Sitcom-star-based clothing retrieval for video advertising: a deep learning framework," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7361–7380, 2019.

[42] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, and X. Xue, "Multi-scale deep learning architectures for person re-identification," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5399–5408, Venice, Italy, October 2017.

[43] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis, "Looking beyond appearances: Synthetic training data for deep CNNs in re- identification," *Computer Vision and Image Understanding*, vol. 167, pp. 50–62, 2018.

[44] W. Wang and U. Neumann, "Depth-aware cnn for rgb-d segmentation," *Proceedings of the European Conference on Computer Vision (ECCV)*, , pp. 135–150, Springer, Cham, 2018.

[45] L. Bobadilla, O. Sanchez, J. Czarnowski, and S. M. LaValle, "Minimalist multiple target tracking using directional sensor beams," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3101–3107, San Francisco, CA, USA, September 2011.

[46] J. Burdick and P. Fiorini, "Minimalist jumping robots for celestial exploration," *International Journal of Robotics Research*, vol. 22, no. 7–8, pp. 653–674, 2003.

[47] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermüller, and Y. Aloimonos, "GapFlyt: active vision based minimalist structure-less gap detection for quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2799–2806, 2018.

[48] C. S. Lynnes, P. G. Macharrie, M. Elkins, T. Joshi, and L. H. Fenichel, "A fast, minimalist search tool for remote sensing data," *AGU Fall Meeting Abstracts*, 2005.

[49] P. Pooj, M. Grossberg, P. Belhumeur, and S. Nayar, "The minimalist camera," *Communications of the ACM*, vol. 60, no. 11, pp. 54–61, 2017.

[50] M. Ltd, "Azure Kinect body tracking joints," Microsoft, 2021, October 2021, https://docs.microsoft.com/en-us/azure/kinect-dk/body-joints.

[51] Cubemos, "Skeleton tracking SDK," Microsoft, 2021, September 2021, https://docs.microsoft.com/en-us/azure/kinect-dk/body-joints.

[52] B. Lewandowski, J. Liebner, T. Wengefeld, S. Miller, and H. Gross, "Fast and robust 3D person detector and posture estimator for mobile robotic applications," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4869–4875, Montreal, QC, Canada, May 2019.

[53] M. Liang and Y. Hu, "Application of human body posture recognition technology in robot platform for nursing empty-nesters," in *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 91–95, Singapore, April 2020.

[54] M. S. Rasoulidanesh and S. Payandeh, "A novel change-detection scheduler for a network of depth sensors," *Journal of Visual Communication and Image Representation*, vol. 66, article 102733, 2020.

[55] W. Quan, H. Li, C. Han et al., "A depth enhancement strategy for kinect depth image," in *MIPPR 2017: Pattern Recognition and Computer Vision*, Xiangyang, China, March 2018.

[56] B. Huhle, T. Schairer, P. Jenke, and W. Strasser, "Robust non-local denoising of colored depth data," in *2008 {IEEE} Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, USA, June 2008.

[57] J. W. Tukey, *Exploratory Data Analysis: Limited Preliminary Ed*, Addison-Wesley Publishing Company, 1970.

[58] S. D. M. Rasouli and S. Payandeh, "Dynamic posture estimation in a network of depth sensors using sample points," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1710–1715, Banff, AB, Canada, October 2017.

[59] M. S. Rasouli D and S. Payandeh, "A novel depth image analysis for sleep posture estimation," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 1999–2014, 2019.

[60] S. D. M. Rasouli and S. Payandeh, "A novel human posture estimation using single depth image from Kinect v2 sensor," in *2018 Annual IEEE International Systems Conference (SysCon)*, pp. 1–7, Vancouver, BC, Canada, April 2018.

[61] W. Zewen and S. Payandeh, "Toward design of a drip-stand patient follower robot," *Journal of Robotics*, vol. 2020, Article ID 9080642, 16 pages, 2020.

[62] V. B. Semwal, S. A. Katiyar, R. Chakraborty, and G. C. Nandi, "Biologically-inspired push recovery capable bipedal locomotion modeling through hybrid automata," *Robotics and Autonomous Systems*, vol. 70, pp. 181–190, 2015.

[63] V. Bijalwan, V. B. Semwal, G. Singh, and T. K. Mandal, "HDL-PSR: Modelling Spatio-Temporal Features Using Hybrid Deep Learning Approach for Post-Stroke Rehabilitation," *Neural Processing Letters*, vol. 54, no. 3, pp. 1–20, 2022.

[64] V. B. Semwal, P. Lalwani, M. K. Mishra, V. Bijalwan, and J. S. Chadha, "An optimized feature selection using bio-geography optimization technique for human walking activities recognition," *Computing*, vol. 103, no. 12, pp. 2893–2914, 2021.

[65] V. B. Semwal, A. Gupta, and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," *The Journal of Supercomputing*, vol. 77, no. 11, pp. 12256–12279, 2021.

[66] A. Gupta and V. B. Semwal, "Occluded gait reconstruction in multi person gait environment using different numerical methods," *Multimedia Tools and Applications*, vol. 81, no. 16, pp. 23421–23448, 2022.

[67] N. Dua, S. N. Singh, V. B. Semwal, and S. K. Challa, "Inception Inspired CNN-GRU Hybrid Network for Human Activity Recognition," *Multimedia Tools and Applications*, pp. 1–35, 2022.

[68] V. B. Semwal, "Data driven computational model for bipedal walking and push recovery," 2017, https://arxiv.org/abs/1710.06548.

[69] V. B. Semwal, A. Bhushan, and G. C. Nandi, "Study of humanoid push recovery based on experiments," in *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, Jabalpur, India, December 2013.